Examen #2

Semana: 10

Nombre del estudiante:
Serlio Alejandro Girón Paz 12141146

Sede de estudio:
UNITEC Tegucigalpa

Docente:
Román Arturo Pineda Soto

Sección:
1741 SISTEMAS OPERATIVOS II 2024Q2

Lugar y fecha de entrega:
Comayagüela M.D.C. 18 de junio de 2024

Universidad Tecnológica Centroamericana

Facultad de Ingeniería

Sistemas Operativos 2 - 6:30 p.m.

25 puntos oro

En el lenguaje de programación de su elección elabore un simulador de caché en disco.

El programa debe generar de forma aleatoria un espacio de direcciones y cargarlos con datos (también pueden ser aleatorios, pero con sentido).

Debe seleccionar un espacio de al menos ⅛ del tamaño del espacio principal de memoria para el caché.

Debe simular dos procesos consumidores del espacio y generar lecturas aleatorias, estas lecturas deben llenar el caché y este se debe reciclar de forma circular.

Si una lectura se encuentra en el caché debe escribir en pantalla "match de caché" y mostrar el valor que se obtuvo del mismo.

Debe considerar todas las buenas prácticas vistas en clase para el buen manejo del caché.

El primer paso es generar paginas de manera aleatoria, esto lo hice con las siguientes funciones.

```
def fill_file():
    with open("disco.txt", "w") as file:
        for i in range(memoria):
            line = f"{i} {generate_string()}\n"
            file.write(line)

def generate_string():
    chars = string.ascii_letters + string.digits
    return ''.join(random.choice(chars) for _ in range(3))
```

Con la función fill_file llenamos un archivo disco donde guardamos páginas. Guardamos el id de la página y guardamos su contenido. El contenido lo generamos con generate_string. Con la función generate_string genero cadenas aleatorias de tres caracteres. La cadena es una cadena alfanumérica. Después de tener el disco con todas las paginas comenzamos con el trabajo de la cache.

```
memoria = 800
capacidad_cache = memoria // 8
cache = []

class MyClass:
    def __init__(self, id, contenido):
        self.id = id
        self.contenido = contenido
```

En este proyecto maneje los dos procesos de manera concurrente con la ayuda de threads. Desde el main creo dos threads y les asignó la función a realizar. De igual manera le envió a los threads cuantas lecturas deben generar con la variable ‘consumir’.

```python
def main():
    # print("Hello, world!")
    fill_file()
    consumir = input("paginas a consumir: ")
    print("cache capacity: ", capacidad_cache, end="\n\n")

    thread1 = threading.Thread(target=buscar_o_cargar_en_cache, args=(int(consumir),1))
    thread2 = threading.Thread(target=buscar_o_cargar_en_cache, args=(int(consumir),2))

    thread1.start()
    thread2.start()

    thread1.join()
    thread2.join()
```

Cada uno de los threads realiza la siguiente tarea.

```python
def buscar_o_cargar_en_cache(cont, identificacion):
    for _ in range(cont):
        id_aleatorio = random.randint(0, memoria - 1)  # Genera un número aleatorio en el rango deseado
        for objeto in cache:
            if objeto.id == id_aleatorio:
                print(f"Thread {identificacion} match de caché: {objeto.contenido}")
                # print_cache()
                break
        # Si no se encuentra en cache, buscar en el archivo
        with open("disco.txt", "r") as file:
            for line in file:
                id_archivo, contenido_archivo = line.split(maxsplit=1)
                if int(id_archivo) == id_aleatorio:
                    # Crear un nuevo objeto MyClass y añadirlo al cache
                    nuevo_objeto = MyClass(id_aleatorio, contenido_archivo.strip())
                    if len(cache) >= capacidad_cache:
                        cache.pop(0)  # Elimina el primer elemento si el cache está lleno
                    cache.append(nuevo_objeto)
                    break
        # print_cache()
```

Esta tarea recibe el numero de lecturas a generar. Primero toma un numero aleatorio dentro del rango de ids que se encuentran en el disco duro. Luego va a la cache y busca si hay se hace match. Si no se hace match añade la página a la cache.

Salidas

Dependiendo del numero de lecturas que le pedimos a los procesos generar se determina el numero de matches, mientras más lecturas pidamos generar. Mas matches tendremos.

```
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2> py .\main.py
paginas a consumir: 10
cache capacity:  100

Thread 1 match de caché: ouF
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2>
```

```
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2> py .\main.py
paginas a consumir: 30
cache capacity:  100

Thread 1 match de caché: 1Rc
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2> py .\main.py
paginas a consumir: 30
cache capacity:  100

Thread 1 match de caché: 9vD
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2> py .\main.py
paginas a consumir: 30
cache capacity:  100

Thread 2 match de caché: SI3
Thread 2 match de caché: CzM
Thread 1 match de caché: 0Ow
Thread 2 match de caché: EMM
Thread 2 match de caché: a2v
Thread 2 match de caché: lSm
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2> py .\main.py
paginas a consumir: 30
cache capacity:  100

Thread 1 match de caché: Aoy
Thread 1 match de caché: KU9
Thread 2 match de caché: 4de
Thread 1 match de caché: jJt
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2> py .\main.py
paginas a consumir: 30
cache capacity:  100

Thread 1 match de caché: 5ZT
Thread 2 match de caché: Edo
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2>
```

Para cada corrida se hacen diferentes matches y diferentes números de matches.

Para tener un numero considerable de matches necesitamos un numero considerable de lecturas generadas.

```
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2> py .\main.py
paginas a consumir: 50
cache capacity:  100

Thread 1 match de caché: GKO
Thread 2 match de caché: adT
Thread 2 match de caché: fsK
Thread 1 match de caché: Xvc
Thread 1 match de caché: N4E
Thread 1 match de caché: i5e
```

```
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2> py .\main.py
paginas a consumir: 100
cache capacity:  100

Thread 2 match de caché: lg5
Thread 1 match de caché: 6O1
Thread 1 match de caché: c9N
Thread 1 match de caché: BLv
Thread 1 match de caché: Ya5
Thread 1 match de caché: ntY
Thread 1 match de caché: lDI
Thread 2 match de caché: OUn
Thread 1 match de caché: QbR
Thread 2 match de caché: My8
Thread 1 match de caché: 7LK
Thread 1 match de caché: wLw
Thread 2 match de caché: xKU
Thread 2 match de caché: ilU
Thread 2 match de caché: SD0
Thread 2 match de caché: X8G
Thread 2 match de caché: eNH
Thread 2 match de caché: m5W
Thread 2 match de caché: cFg
Thread 2 match de caché: f7E
Thread 2 match de caché: m5W
PS C:\Users\serli\OneDrive - Universidad Tecnologica Centroamericana\sistemas operativos\operativos II\Examen 2>
Ln 65, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  3.12.0 64-bit  Prettie
```

```
paginas a consumir: 200
cache capacity:  100

Thread 2 match de caché: 7au
Thread 1 match de caché: fum
Thread 2 match de caché: XKq
Thread 2 match de caché: CEN
Thread 1 match de caché: 6Av
Thread 2 match de caché: Da9
Thread 1 match de caché: gFp
Thread 1 match de caché: gFp
Thread 1 match de caché: 14r
Thread 1 match de caché: 7B2
Thread 2 match de caché: dx0
Thread 2 match de caché: 14r
Thread 1 match de caché: GBA
Thread 1 match de caché: bwf
Thread 1 match de caché: P1s
Thread 1 match de caché: jiw
Thread 2 match de caché: LOW
Thread 2 match de caché: 34n
Thread 2 match de caché: dv5
Thread 2 match de caché: vvW
Thread 1 match de caché: Ml2
Thread 2 match de caché: QVl
Thread 1 match de caché: bwf
Thread 1 match de caché: gFp
Thread 1 match de caché: se2
Thread 2 match de caché: HRR
Thread 2 match de caché: 4sF
Thread 1 match de caché: 8Ke
Thread 2 match de caché: 5au
Thread 1 match de caché: IBD
Thread 2 match de caché: gFp
Thread 2 match de caché: d4v
```

```
PS C:\Users\serli\OneDrive - Universidac
paginas a consumir: 300
cache capacity:  100

Thread 1 match de caché: 9EL
Thread 1 match de caché: 5SM
Thread 1 match de caché: wSk
Thread 1 match de caché: WZo
Thread 1 match de caché: IyH
Thread 2 match de caché: Z6T
Thread 1 match de caché: kuT
Thread 2 match de caché: qDN
Thread 1 match de caché: MQS
Thread 2 match de caché: QDd
Thread 1 match de caché: Li4
Thread 2 match de caché: Zsv
Thread 1 match de caché: JbM
Thread 2 match de caché: 2QY
Thread 1 match de caché: DS3
Thread 1 match de caché: FVd
Thread 2 match de caché: DS3
Thread 2 match de caché: O7t
Thread 2 match de caché: 1Us
Thread 2 match de caché: qDN
Thread 1 match de caché: YBj
Thread 1 match de caché: 2QY
Thread 1 match de caché: V8n
Thread 1 match de caché: hmo
Thread 1 match de caché: S0a
Thread 2 match de caché: sGr
Thread 2 match de caché: pet
Thread 1 match de caché: xdv
Thread 2 match de caché: YDP
Thread 1 match de caché: DEM
Thread 1 match de caché: xdv
Thread 2 match de caché: YDP
Thread 1 match de caché: DEM
Thread 1 match de caché: rla
Thread 1 match de caché: sGr
Thread 1 match de caché: Hqk
Thread 1 match de caché: FBr
Thread 2 match de caché: fND
Thread 1 match de caché: xzK
Thread 2 match de caché: DEM
Thread 2 match de caché: J3D
Thread 1 match de caché: zZ9
Thread 2 match de caché: QUw
Thread 2 match de caché: x42
Thread 2 match de caché: 2QY
Thread 1 match de caché: wt9
Thread 1 match de caché: HsQ
Thread 2 match de caché: OnR
Thread 1 match de caché: Lrq
Thread 2 match de caché: b8b
Thread 2 match de caché: J3D
Thread 1 match de caché: LlW
Thread 2 match de caché: RBs
Thread 1 match de caché: CXm
Thread 1 match de caché: emd
Thread 1 match de caché: 9Uz
Thread 2 match de caché: VPW
Thread 1 match de caché: ruY
Thread 1 match de caché: D5L
Thread 2 match de caché: caT
Thread 2 match de caché: lon
Thread 1 match de caché: zNQ
Thread 1 match de caché: Oro
Thread 2 match de caché: PWI
PS C:\Users\serli\OneDrive - Un
```